	L #	Hits	Search Text	DBs
1	L1	14035	(scatter$3 gather$3) near10 (bit byte element item)	USPAT; US-PGPUB
2	L2	144	1 near20 mask$3	USPAT; US-PGPUB
3	L3	5584	(scatter$3 gather$3) near10 (bit byte element item)	EPO; JPO; DERWENT
4	L4	44	3 near20 mask$3	EPO; JPO; DERWENT; IBM_TDB

FIG. 49, it is provided with a moving object list generating unit 133. The animation data generating unit 16 shown in FIG. 49 represents a natural movement of images on a screen by selecting simultaneously calculated objects as they are scattered uniformly on the screen. The moving object list generating unit 133 rearranges the moving object list 102 such that objects located in distant places on the screen can be positioned adjacently in the screen.

FIG. 50 is a flowchart of the process performed by the moving object list generating unit 133. FIG. 51 shows an example of a moving object list generated through the process activated by the control unit 113 at the start of the animation. An object O_1 on the top of the object tree shown in FIG. 51 is named a route. First, it is checked how many children the route has (step S151). Then, the object list is divided by the number of the children into partial trees (step S152). At this time, the moving objects having animation data are extracted from among the objects in a partial tree starting with the first child. Thus, a moving object list of the first child can be generated. Likewise, the moving object lists of the partial trees can be generated for the second and other children. In the case shown in FIG. 51, since route O_1 has two children O_2 and O_3, the moving object list is divided into two object lists, that is, one for the left object tree and the other for the right object tree. In this case, two lists are generated. They are a moving object list 141 of child O_2 and a moving object list 142 of child O_3. The moving object list 141 of child O_2 contains objects O_2, O_5, and O_7. The moving object list 142 of child O_3 contains objects O_6 and O_8. The child O_3 is, for example, a static object and is not contained in the moving object list 142. Likewise, child O_4 is not contained in the moving object list 141.

Then, a plurality of divided lists are merged into a single moving object list (step S153). The list is generated such that the objects in the plural lists are entered alternately. In the case shown in FIG. 51, the two moving object lists 141 and 142 are merged into a single moving object list 143.

Thus, in the case shown in FIG. 51, generated is a moving object list 143 alternately pointing to an object in the right object tree and an object in the left object tree.

The data acquiring unit 111 in the animation data generating unit 16 shown in FIG. 49 sequentially reads object data through the generated moving object list. Accordingly, in the case shown in FIG. 51, the data of an object in the right and left object trees are alternately retrieved. Thus, the simultaneously rewritten (updated) objects can be adjusted.

Since position information of an object can also be inherited in the structure of an object tree, objects having the same parent are positioned closely. Therefore, regenerating a moving object list as described above controls the generation of display data such that simultaneously calculated objects can be uniformly scattered on a screen.

FIG. 52 shows another example of the configuration of the third embodiment. As shown in FIG. 52, an image editing unit 151 is connected to the animation data generating unit 16 and the update control unit 17. Providing the image editing unit 151 enables a user to change update control data and weighted data.

As described above, since the number of image displays is increased and the images can be displayed at a high speed according to the third embodiment, smoother, easier, and more realistic animation can be successfully realized.

FIG. 53 is a block diagram showing the fourth embodiment of the present invention. The CG data display device according to the fourth embodiment is provided in the graphic display device for displaying CG data, that is, graphic information generated by a computer system.

The first processing unit 18 comprises a CG data storage unit 162 and a CG data management unit 163, manages CG data, and, if a change has arisen in the CG data, outputs as change data the CG data relating to the change.

The second processing unit 23 comprises a changed CG data storage unit 164 and a CG data image generating unit 165, stores CG data to be displayed and generates image data using the input change data.

A change data buffer 161 is provided between the first processing unit 18 and the second processing unit 23, and stores the change data.

For example, the first processor 18 and the second processing unit 23 are processor systems and the change data buffer 161 is a shared memory which can be accessed by the processor systems.

The CG data storage unit 162 stores CG data.

The CG data management unit 163 receives external information, calculates its influence on CG data stored in the CG data storage unit 162, and changes CG data to be changed.

The changed CG data storage unit 164 stores CG data to be displayed.

The CG data image generating unit 165 updates CG data stored in the changed CG data storage unit 164 based on the change data, and generates image data from the updated CG data.

When external information associated with display is provided for the first processing unit 18, the CG data management unit 163 in the first processing unit 18 calculates its influence on CG data stored in the CG data storage unit 162, and obtains changed CG data, that is, change data. Then, the CG data management unit 163 in the first processing unit 18 stores the change data in the change data buffer 161. It also stores the change data in the CG data storage unit 162.

The change data stored in the change data buffer 161 are read by the CG data image generating unit 165 in the second processing unit 23. The CG data image generating unit 165 updates the CG data stored in the changed CG data storage unit 164 based on the change data, and generates image data from the updated CG data.

Since CG data are stored in the first processing unit 18 and the second processing unit 23, the CG data can be changed or updated by transmitting only the change data from the first processing unit 18 to the second processing unit 23, and the amount of information to be transmitted can be reduced with improved throughput at a high speed.

Practically, CG data are composed of form data and viewpoint data, for example. They are stored in the first processing unit 18 and the second processing unit 23. If the angle of the viewpoint data is changed from 15° to 45° by 30°, the change data indicating that the viewpoint data have changed into 45° are transmitted from the first processing unit 18 to the second processing unit 23. In the second processing unit 23, form data and viewpoint data 45° are stored in changed CG data storage unit 164, and images are generated from the CG data based on the stored data. The data transmission time can be considerably reduced because all CG data in a displayed world are not transmitted but only data relating that the viewpoint data changed into 45°, that is, the viewpoint data 45°, are transmitted from the first processing unit 18 to the second processing unit 23. As a result, both form data and viewpoint data 45° are stored in the first processing unit 18 and the second processing unit 23. In a multiprocessor system, the first processing unit 18 detects that viewpoint data are changed from 15° to 45°, and concurrently the second processing unit 23 generates an image.

	Document ID	U	Title	Current OR
1	US 20040047014 A1	☐	In-line holographic mask for micromachining	359/15
2	US 20040038135 A1	☒	Photolithographic mask and methods for producing a structure and of exposing a wafer in a projection apparatus	430/5
3	US 20040025733 A1	☒	EUV lithographic projection apparatus comprising an optical element with a self-assembled monolayer, optical element with a self-assembled monolayer, method of applying a self-assembled monolayer, device manufacturing method and device manufactured thereby	101/494
4	US 20040023130 A1	☒	Test photomask, flare evaluation method, and flare compensation method	430/5
5	US 20040017568 A1	☒	Absolute measurement centrifuge	356/338
6	US 20040016686 A1	☒	Absolute measurement centrifuge	210/94
7	US 20040011975 A1	☒	Sensors and methods for high-sensitivity optical particle counting and sizing	250/574
8	US 20030235265 A1	☒	High spatial resolution X-ray computed tomography (CT) system	378/4
9	US 20030197690 A1	☒	Computer input system	345/179
10	US 20030171908 A1	☒	Simulation and timing control for hardware accelerated simulation	703/16
11	US 20030155532 A1	☒	Electron-beam lithography	250/492.3
12	US 20030152846 A1	☒	Photolithographic mask	430/5
13	US 20030107720 A1	☒	Continuously adjustable neutral density area filter	355/77
14	US 20030091911 A1	☒	Photolithography mask and method of fabricating a photolithography mask	430/5
15	US 20030077521 A1	☒	Method for producing scatter lines in mask structures for fabricating integrated electrical circuits	430/5
16	US 20030034445 A1	☒	Light guide for use with backlit display	250/227.11
17	US 20030022074 A1	☒	Photolithographic mask	430/5

FIG. 54 shows a practical configuration according to the fourth embodiment. The CG data display device shown in FIG. 54 comprises a first processor 170 and a second processor 180.

The first processor 170 comprises a CPU 171, a read only memory (ROM) 172, a working random access memory (RAM) 173, an input/output processor (IOP) 174, a CG data RAM 175, and a shared memory interface (IF) 176. These circuits are commonly connected to one another via a CPU bus 177.

The CPU 171 executes a program stored in the ROM 72, receives display information from the IOP 174 such as CG data, CG information from an operator, etc., and determines whether or not the currently displayed information should be changed. The working RAM 173 is a working area used when the CPU 171 executes a program in the ROM 172. Additionally, this system is provided with the CG data RAM 175. The CG data RAM 175 stores CG data to be displayed. As described later, the CG data stored in the CG data RAM 175 are also stored in a RAM 185 in the second processor 180.

In the first processor 170, if CG data have been changed, that is, change data have arisen, the first processor 170 stores the change data in a shared memory 178 through the shared memory interface (IF) 176, and stores the change data in the CG data RAM 175. Then, the change data stored in the shared memory 178 are read by the second processor 180 to change the image which has been generated and displayed.

The second processor 180 comprises a CPU 181, a ROM 182, a working RAM 183, a video RAM 184, a changed CG data RAM 185, and a shared memory interface (IF) 186. These circuits are commonly connected to a bus 187 of the CPU 181. The CPU 181 of the second processor 180 executes a program stored in the ROM 182 and generates images using the working RAM 183. The CG data used in the generation of images (that is, changed CG data) are stored in the changed CG data RAM 185.

If change data are stored in the shared memory 178 by the first processor 170, the second processor 180 reads the change data stored in the shared memory 178 through the shared memory interface (IF) 186, and adds them to the CG data in the changed CG data RAM 185. Then, the second processor 180 generates dot data to be displayed from the changed CG data, and writes them into the video RAM 184. The video RAM 184 is a circuit for generating a video signal from the dot data. The video signal is added to, for example, a cathode ray tube (CRT) not shown in the attached drawings and is displayed on the CRT.

CG data are stored in the above described CG data RAM 175 and the changed CG data RAM 185. The storage format can be the same or different in these RAMS. For example, the CG data RAM 175 in the first processor 170 can store the CG data in a format in which a user can easily give his or her instruction (for example, a tree structure). The changed CG data RAM 185 in the second processor 180 can store the data in a format in which images are easily generated (for example, a parallel arrangement in item units).

As described above, the first processor 170 and the second processor 180 separately store CG data, and the first processor 170 transmits only change data and the second processor 180 displays CG data after updating them. Therefore, all CG data are not required to be transmitted in association with a change, thereby performing the process at a high speed. Furthermore, parallel processing can speed up the entire process.

A processor is used in the above described embodiment, but the present invention is not limited to this application. For example, each unit can be structured by hardware other than a processor or by a process of a computer system.

The block diagram shown in FIG. 53 is explained below in detail. FIG. 55 shows the configuration of the CG data management unit 163 shown in FIG. 53. The CG data management unit 163 comprises an input interpreting unit 191, a CG calculating unit 192, a CG data referring unit 193, a change data storing unit 194, and a CG data storing unit 195, where solid-line arrows in FIG. 55 indicate the direction of control and the flow of data, and broken-line arrows show the flow of data.

The input interpreting unit 191 receives an input from a mouse and a keyboard, interprets the meaning of the input information, and instructs the CG calculating unit 192 to perform an calculation. In interpreting the information, it also determines what sort of calculation should be performed and informs the CG calculating unit 192 of the determination. The CG calculating unit 192 requests the CG data referring unit 193 to refer to necessary data, and performs a specified calculation on obtained data. Then, it outputs the calculation result to the CG data storing unit 195. Then, the output data are also transmitted to the change data storing unit 194 as change data, and instructs it to store them into the change data buffer 161.

The CG data referring unit 193 retrieves data stored in the CG data storage unit 162 upon receipt of a data retrieval request, and notifies the CG calculating unit 192 of the result. The CG data storing unit 195 stores CG data in the CG data storage unit 162. The change data storing unit 194 stores change data in the change data buffer 161.

Each unit of the CG data management unit 163 is operated as follows. First, upon receipt of data from a keyboard, the input interpreting unit 191 interprets the contents of the input information, and outputs necessary information such as a calculation method, etc. to the CG calculating unit 192. The CG calculating unit 192 requests the CG data referring unit 193 to output data corresponding to the specified calculation method. The CG data referring unit 193 reads corresponding data stored in the CG data storage unit 162, and outputs them to the CG calculating unit 192. The CG calculating unit 192 performs a calculating process using the data and obtains all or a part of the changed CG data as change data. Then, it transmits the CG data to be storing to the CG data storing unit 195.

The CG data storing unit 195 stores the received CG data in the CG data storing unit 162. Thus, the CG data include a change arising during a display process and then are stored in the CG data storage unit 162. Furthermore, to keep the consistency of the information stored in the first processing unit 18 and the second processing unit 23, the CG data storing unit 195 instructs the change data storing unit 194 to store the change data in the change data buffer 161. Then, the change data storing unit 194 stores the change data in the change data buffer 161. Thus, the CG data management unit 163 obtains CG data relating to the input change, and stores it in the CG data storage unit 162 and the change data buffer 161.

In the above described process, data are input from, for example, a mouse and a keyboard. However, the present invention is not limited to these input units, and can receive a display request from, for example, an external computer system. Displayed CG data stored in a ROM can be stored in a RAM at initialization if the information to be displayed is preliminarily limited. Furthermore, they can be added from an external computer system at initialization as initialization information, and then be displayed.

FIG. 56 shows the configuration of the CG data image generating unit 165. The CG data image generating unit 165

	Document ID	U	Title	Current OR
18	US 20020154018 A1	☒	Fire detector unit	340/630
19	US 20020149691 A1	☒	Aperture coded camera for three dimensional imaging	348/335
20	US 20020116602 A1	☒	Partial bitwise permutations	712/223
21	US 20020057276 A1	☒	Data processing apparatus, processor and control method	345/555
22	US 20020048708 A1	☒	Method of patterning sub-0.25lambda line features with high transmission, "attenuated" phase shift masks	430/5
23	US 20020039209 A1	☒	IN-LINE HOLOGRAPHIC MASK FOR MICROMACHINING	359/15
24	US 20020033952 A1	☒	Control of position and orientation of sub-wavelength aperture array in near-field microscopy	356/512
25	US 20020027974 A1	☒	X-RAY EXPOSURE METHOD INCLUDING M-SHELL AND/OR L-SHELL ABSORPTION EDGES AT PREDETERMINED WAVELENGTHS	378/145
26	US 20020024011 A1	☒	Method for correcting opaque defects in reticles for charged-particle-beam microlithography, and reticles produced using same	250/307
27	US 20020021492 A1	☒	Stereoscopic image display method and stereoscopic image display apparatus using it	359/463
28	US 20020021451 A1	☒	Scanning interferometric near-field confocal microscopy with background amplitude reduction and compensation	356/511
29	US 20020016581 A1	☒	Absorbent article with improved surface fastening system	604/386
30	US 20010031404 A1	☒	Process for fabricating a projection electron lithography mask and a removable, reusable cover for use therein	430/5
31	US 20010021477 A1	☒	Method of manufacturing a device by means of a mask phase-shifiting mask for use in said method	430/5
32	US 20010014936 A1	☒	Data processing device, system, and method using a table	711/221
33	US 6724462 B1	☒	Capping layer for EUV optical elements	355/53
34	US 6687801 B1	☒	Adaptive copy pending off mode	711/162
35	US 6667809 B2	☒	Scanning interferometric near-field confocal microscopy with background amplitude reduction and compensation	356/511

forming part of the second processing unit **23** comprises a changed CG data referring unit **201**, an image generating unit **202**, a change data referring unit **203**, and a changed CG data storing unit **204**, where solid-line arrows in FIG. **55** indicate the direction of control and the flow of data, and broken-line arrows show the flow of data.

The image generating unit **202** transmits change data to the change data referring unit **203**, instructs the change data referring unit **203** to update changed CG data, obtains the changed CG data from the changed CG data referring unit **201**, and generates images. The change data referring unit **203** refers to the contents, corresponding to the change data, of the changed CG data storage unit **164**, and transmits the change data to the changed CG data storing unit **204** if a change is required. The changed CG data storing unit **204** updates the changed CG data stored in the changed CG data storage unit **164** according to the contents given by the change data referring unit **203**. The changed CG data referring unit **201** reads required CG data from the changed CG data storage unit **164**.

The CG data image generating unit **165** shown in FIG. **56** operates as follows. When the change data stored in the change data buffer **161** are received by the image generating unit **202**, the image generating unit **202** requests the change data referring unit **203** to update changed CG data.

If there are change data, the change data referring unit **203** outputs their contents to the changed CG data storing unit **204**, and instructs the changed CG data storing unit **204** to update the changed CG data. The changed CG data storing unit **204** partially updates the changed CG data stored in the changed CG data storage unit **164** based on the received contents of the change data.

The image generating unit **202** requests the changed CG data referring unit **201** to read CG data required to change an image, and generates image from the updated CG data returned from the changed CG data referring unit **201**. The read requests are issued almost simultaneously to the change data referring unit **203** and the changed CG data referring unit **201**. Furthermore, the update of changed CG data, the read of updated CG data, and the generation of images are carried out simultaneously.

The CG data management unit **163** and the CG data image generating unit **165** are operated as described above.

According to the present invention, CG data are stored separately in the CG data storage unit **162** and the changed CG data storage unit **164**. Accordingly, the information transmitted through the change data buffer **161** can be change data only, and the amount of the information can be considerably reduced, thereby taking shorter time for transmission and resulting in a high speed process. Since, for example, change information provided continuously for the device shown in FIG. **53** can be sequentially processed in parallel in a pipeline system, the entire process can be completed more quickly.

According to the fourth embodiment, the first processing unit **18** and the second processing unit **23** perform respective processes in the units shown in FIGS. **53**, **55**, and **56**. If the processes are performed by processors, then the process of each unit shown in FIGS. **53**, **55**, and **56** is performed corresponding to each of the processes shown in FIGS. **57**, **58**, **59**, **60**, and **61** performed by a processor. These processes can be also performed in multiple processes (multi-process) defined in a computer system. If each process of the first processing unit **18** and the second processing unit **23** is performed by a process of the multiple processes, then the change data only can be transmitted between processes because each process stores CG data. The transmission can be performed within a short time and the entire process can be completed quickly.

FIG. **57** shows the operation of the first processing unit **18**. The first processing unit **18** constantly performs a CG data management process in step S161. That is, at the start of its operation, it performs the CG data management process in step S161. If the CG data management process is completed, it is determined whether or not the entire process has been completed in step S162. If no, the CG data management process is performed again in step S161. Thus, the CG data management process is constantly carried out.

The CG data management process in step S161 is composed of an input interpreting process in step S171 and a CG calculating process in step S172 as shown in FIG. **58**. When the CG data management process is started, the input interpreting process is carried out in step S171. The input interpreting process in step S171 is composed of the processes in steps S173 through S175. First, data are input in step S173. The input is made through, for example, a mouse. Then, a function is selected in step S174 and the amount of a change is calculated in step S175 based on the input information. Interpreting is a selection of a function in step S174. For example, moving a mouse is interpreted as movement of a viewpoint. Calculating the amount of a change using the input data in step S175 is a process of obtaining data used in the following calculation, that is, data indicating that the mouse has moved 3 dots to the right.

If the input interpreting process in step S171 has been completed, the CG calculating process in step S172 is started. The CG calculating process in step S172 is composed of the processes in steps 176 through 179. First, in step S176, the CG data referring process is performed, and the CG data required in a calculation are referred to. In step S177, the CG data changed (change data) are obtained according to the function selected in step S174, the amount of a change obtained in step S175 and the present CG data. That is, obtained is the result that moving the mouse 3 dots to the right has made a change of 30° from the original viewpoint of 15° to a viewpoint of 45°. The change data are stored in the CG data storage unit **162** in the CG data storing process in step S178. Then, the change data are stored in the change data buffer **161** in the change data storing process in step S179.

FIG. **59** shows further in detail the above described CG calculating process in step S172. First in the CG calculating process, CG data referring process is performed in step S176. That is, CG data are retrieved according to the function in step S181. Then, the retrieved CG data and their index are returned in step S182, and the CG data are changed in step S177 according to the function and the amount of the change. Then, in the CG data storing process in step S178, the CG data changed are stored according to the index in the CG data storage unit **162**. In the change data storing process in step S179, the change data of CG data are stored in the change data buffer **161**.

As described above, the first processing unit **18** starts its operation and stores change data in the change data buffer **161**. If the change data buffer has stored the change data, the second processing unit **23** starts its operation.

FIG. **60** shows the operation of the second processing unit **23**. The CG data image generating unit **165** of the second processing unit **23** performs a CG data image generating process in step S191 in which image data representing an image to be displayed are generated from CG data. If the CG data image generating process has been completed, it is determined whether or not it has been actually completed. If no, the CG data image generating process is performed again in step S191.

	Document ID	U	Title	Current OR
36	US 6631369 B1	☒	Method and system for incremental web crawling	707/4
37	US 6618174 B2	☒	In-line holographic mask for micromachining	359/15
38	US 6576887 B2	☒	Light guide for use with backlit display	250/227.11
39	US 6552805 B2	☒	Control of position and orientation of sub-wavelength aperture array in near-field microscopy	356/511
40	US 6549959 B1	☒	Detecting modification to computer memory by a DMA device	710/22
41	US 6544694 B2	☒	Method of manufacturing a device by means of a mask phase-shifting mask for use in said method	430/5
42	US 6510201 B2	☒	Apparatus for measuring the pulse transmission spectrum of elastically scattered x-ray quantities	378/86
43	US 6498351 B1	☒	Illumination system for shaping extreme ultraviolet radiation used in a lithographic projection apparatus	250/492.2
44	US 6482555 B2	☒	Method of patterning sub-0.25.lambda. line features with high transmission, "attenuated" phase shift masks	430/5
45	US 6468700 B1	☒	Transfer mask blanks and transfer masks exhibiting reduced distortion, and methods for making same	430/5
46	US 6448569 B1	☒	Bonded article having improved crystalline structure and work function uniformity and method for making the same	250/493.1
47	US 6444398 B1	☒	Method for manufacturing a semiconductor wafer using a mask that has several regions with different scattering ability	430/296
48	US 6397379 B1	☒	Recording in a program execution profile references to a memory-mapped active device	717/140
49	US 6381300 B1	☒	Exposure mask, exposure mask manufacturing method, and semiconductor device manufacturing method using exposure mask	378/35
50	US 6377726 B1	☒	Transverse mode transformer	385/28
51	US 6372393 B2	☒	Process for fabricating a projection electron lithography mask and a removable, reusable cover for use therein	430/5
52	US 6366639 B1	☒	X-ray mask, method of manufacturing the same, and X-ray exposure method	378/34
53	US 6355384 B1	☒	Mask, its method of formation, and a semiconductor device made thereby	430/5
54	US 6330333 B1	☒	Cryptographic system for wireless communications	380/207
55	US 6312854 B1	☒	Method of patterning sub-0.25 lambda line features with high transmission, "attenuated" phase shift masks	430/5
56	US 6297169 B1	☒	Method for forming a semiconductor device using a mask having a self-assembled monolayer	438/736
57	US 6278847 B1	☒	Aperture coded camera for three dimensional imaging	396/324
58	US 6261726 B1	☒	Projection electron-beam lithography masks using advanced materials and membrane size	430/5

FIG. 61 shows the above described CG data image generating process in step S191. When the process is started, the change data referring process is performed in step S201. First, in the change data referring process, it is determined in step S205 whether or not there are change data. If yes, a changed CG data storing process in step S206 is performed to store according to the index the changed CG data updated based on the change data. Then, it is determined again whether or not there are any change data in step S205.

If there are no change data (no in step S205), a changed CG data referring process is performed in step S202, and CG data are referred to according to an index, and are retrieved. Then, in step S203, the CG data referred to, that is, the read CG data are converted into an image. In step S204, it is determined whether or not all data have been referred to. If no, the processes are repeatedly performed again from step S202. If all data have been referred to, the CG data image generating process in step S191 terminates.

As described above, input images are managed by the first processing unit 18, and a changing process is performed based on change data and an image is generated simultaneously by the second processing unit 23. If two processors are used to manage the change data and to generate the image changed, they are operated in parallel. Parallel processing can be performed because the first processing unit 18 and the second processing unit 23 are provided with the CG data storage unit 162 and the changed CG data storage unit 164 respectively. Furthermore, only the data associated with a change are transmitted between the two processing units, the amount of the transmission data can be reduced, thereby taking a shorter time for transmission. That is, even if a large amount of CG data are changed at very short intervals, the CG data management unit 163 for managing CG data and the CG data image generating unit 165 for processing data changes are separately operated asynchronously at a high speed. Only transmitting a data portion directly changed reduces the amount of transmission data, thereby further speeding up the entire process.

In a process of a computer system operating with multi-process mechanism, an asynchronous operation realizes quick response of the system. Therefore, even a single computer system can perform the above described process at a high speed.

FIG. 62 shows the processes performed in parallel by the first processing unit 18 and the second processing unit 23. In FIG. 62, while a changed screen is generated based on the input by the (n−1)th operation in the CG data image generating process of the second processing unit 23, the n-th input is provided for the first processing unit 18.

At this time, the first processing unit 18 performs the n-th input interpreting process followed by the CG calculating process in parallel to the (n−1)th CG data image generating process of the second processing unit 23. If the (n−1)th CG data image generating process has been completed, the second processing unit 23 performs the n-th CG data image generating process based on the n-th change data received from the first processing unit 18.

The first processing unit 18 and the second processing unit 23 also process the (n+1)th and the (n+2)th inputs in parallel, and sequentially display CG data.

FIGS. 63 through 65 show more practical examples of the fourth embodiment. FIG. 63 shows the configuration of a flight simulator. FIG. 64 shows the configuration of a test course simulator. FIG. 65 shows the configuration of a scene simulator.

The flight simulator in FIGS. 63 simulates a flight of an airplane, and is provided with a CG data display device 210. A pseudo flying device 211 comprises a user-operated handle, pedal, etc. which are input units of the simulator of flying the airplane. A plane movement calculation mechanism 212 calculates the present state of the plane body. For example, it calculates the state of the airplane as it is when the airplane is taking off or landing. A training course execution mechanism 213 instructs an operator to fly the airplane in accordance with the present training course if the present embodiment is a training simulator.

The information obtained by the above described device and mechanisms is input to an operation integrating mechanism 214. The operation integrating mechanism 214 generates a view from the present pilot seat and outputs to the CG data display device 210. Display data, that is, CG data, of various information to be displayed like a layout of an airport, a state of clouds, etc. are stored in the operation integrating mechanism 214 and output to the CG data display device 210 at any time when necessary. The CG data display device 210 has the configuration as shown in FIG. 53. Unless a new unit of display is specified, display images are generated according to the input information, for example, a specified direction, etc., and are displayed on a CRT or other display devices.

The test course simulator shown in FIG. 64 simulates the test course for an automobile. A road surface designing device 221 generates coordinate data of the surface of a course road including, for example, a bank obtained through a computer aided design (CAD), etc. The information generated by the road surface designing device 221 is input as road surface form data 222 to a run regenerating device 225. The road surface form data 222 are input to a run state calculating mechanism 223. The run state calculating mechanism 223 calculates and outputs run data in a specified run state. The run state calculating mechanism 223 generates run state data 224.

The run regenerating device 225 operates according to road surface form data 222 and run state data 224. Screen information at, for example, a bank is generated when a test run is performed, and is output to the CG data display device 210. The run regenerating device 225 preliminarily stores display information, that is, CG data in the CG data display device 210, moves a viewpoint according to the result of the run state, and instructs the CG data display device 210 to display an image from the viewpoint. Thus, a test course simulation can be realized.

FIG. 65 shows the configuration of a scene simulator. Form data 232 are generated by a form generating CAD mechanism 231. If a viewpoint data input mechanism 233 enters viewpoint data from which a scene is viewed, a scene generating mechanism 234 generates a scene as being viewed from the viewpoint, and outputs the scene to the CG data display device 210. For example, if a tower, for example, an iron tower is to be constructed, the system operates as follows. To evaluate the difference between the scene without a tower and the scene with the tower to be constructed, the CG data display device 210 preliminarily stores and displays the scene without the tower. If the form data 232 generated by the form generating CAD mechanism 231 are input as information about the tower, then the scene generating mechanism 234 inputs to the CG data display device 210 the changed portion display information, that is, the CG data of the tower, and the position information about the viewpoint of the scene. Then, the CG data display device 210 displays the scene with the tower. Even though the viewpoint has moved, the CG data display device 210 does not change all CG data, but changes only a necessary portion of the CG data and performs in parallel a display process

	Document ID	U	Title	Current OR
59	US 6251543 B1	☒	Process for fabricating a projection electron lithography mask and a removable reusable cover for use therein	430/5
60	US 6246451 B1	☒	Stereoscopic image displaying method and stereoscopic image apparatus	349/15
61	US 6151418 A	☒	Method for imaging an area of investigation	382/274
62	US 6127211 A	☒	Method of manufacturing transistor	438/158
63	US 6088545 A	☒	Real-image type viewfinder	396/373
64	US 5990498 A	☒	Light-emitting diode having uniform irradiance distribution	257/99
65	US 5989760 A	☒	Method of processing a substrate utilizing specific chuck	430/22
66	US 5986742 A	☒	Lithographic scanning exposure projection apparatus	355/53
67	US 5936729 A	☒	Photo detector assembly for measuring particle sizes	356/336
68	US 5906902 A	☒	Manufacturing system error detection	430/30
69	US 5889580 A	☒	Scanning-slit exposure device	355/67
70	US 5877858 A	☒	Textured surface monitoring and control apparatus	356/496
71	US 5876881 A	☒	Manufacturing method for mask for charged-particle-beam transfer or mask for x-ray transfer	430/5
72	US 5866913 A	☒	Proximity correction dose modulation for E-beam projection lithography	250/492.22
73	US 5838433 A	☒	Apparatus for detecting defects on a mask	356/364
74	US 5831274 A	☒	Apparatus for image transfer with charged particle beam, and deflector and mask used with such apparatus	250/492.23
75	US 5830064 A	☒	Apparatus and method for distinguishing events which collectively exceed chance expectations and thereby controlling an output	463/22
76	US 5818337 A	☒	Masked passive infrared intrusion detection device and method of operation therefore	340/567
77	US 5798194 A	☒	Masks for charged-particle beam microlithography	430/5
78	US 5789119 A	☒	Image transfer mask for charged particle-beam	430/5
79	US 5773838 A	☒	Apparatus for image transfer with charged particle beam, and deflector and mask used with such apparatus	250/492.23
80	US 5773837 A	☒	Apparatus for image transfer with charged particle beam, and deflector and mask used with such apparatus	250/492.23
81	US 5770180 A	☒	Bridge-substituted tropanes for methods of imaging and therapy	424/1.81

associated with the change. Thus, the data can be displayed at a high speed.

The CG data management unit 163 according to the fourth embodiment is designed to receives an input through a program as well as an input by, for example, a mouse. The scene generating mechanism 234 can be provided with a program for moving a viewpoint along a predetermined course and generating a scene without any operation by a user. According to the information from the program, the CG data display device 210 generates and displays image data. That is, it operates as if viewpoint data were constantly received from the viewpoint data input mechanism 233.

The instruction from the program is not limited to the application in the scene simulator shown in FIG. 65, but can be used in a flight simulator and test course simulator. If these systems are the computer systems comprising a multi-process mechanism, then the CG data management unit 163 waits for a call for a function instead of monitoring the operation of a mouse.

To explain the fourth embodiment further in detail, a CG data display device applied to a drive simulator such as a test course simulator, etc. is more concretely explained below. FIG. 66A shows an example of a display screen of a drive simulator provided with the CG data display device 210. In FIG. 66A, the displayed three-dimensional world is represented by an xyz coordinate system, where the y axis is extended horizontally to the right, the z axis is extended vertically upward, and the x axis is extended inward in a vertical direction to the screen. Displayed on the screen shown in FIG. 66A is a scene from a driver's viewpoint toward the moving direction of the car. Moving a mouse to left and right corresponds to operating a handle. The distance between the center of the screen and the mouse pointer is set as the turn angle of the handle.

Assume that a user moves the mouse button 3 picture elements (3 dots) to the right from the center of the screen. In this case, the following determination is made.

A mechanism for monitoring the movement of a mouse is designed to cover the movement in a specified area of the X-WINDOW SYSTEM of UNIX, that is, control is passed to a specified function when the mouse is moved in a window of the X-WINDOW. If control is passed to the function, it is determined that the moving direction of the viewpoint has been changed.

If the mouse is moved, then control is passed to the specified function, and the moving distance and direction of the mouse are calculated based on the position of the mouse after the movement and the position of the mouse previously stored in a memory. The previous position is stored as the position of the mouse in the memory.

Assuming that the mouse has moved to the right the mouse has moved by a 3-picture-element distance, the input interpreting unit 191 of the CG data management unit 163 determines that the direction of the viewpoint has changed to the right with the change amount of 3 picture elements.

With the data, the CG calculating unit 192 requests the CG data referring unit 193 to refer to the present viewpoint data and the viewpoint movement data to perform a calculating process. That is, it provides a retrieval function with a viewpoint identifier and receives a pointer to a structure of a viewpoint. The CG data referring unit 193 retrieves the CG data in the CG data storage unit 162 and obtains a three-dimensional coordinate, a direction vector and a speed of the movement at the position.

The data of the fourth embodiment form a tree structure. FIG. 67 shows a data structure according to the C language. FIG. 68 is a flowchart of retrieving in an object tree.

The identifier of the above described retrieval, that is, a retrieval key, is an identifier of the viewpoint associated with the change request. Based on the identifier, the retrieval is performed as follows. First, the retrieval process starts with a retrieval start object (root object) (step S211). Then, it is determined whether or not the retrieval key and the identifier of the object match each other (step S212). If yes, a pointer to data of the object is returned and the retrieval process terminates (step S217). If they don't match, it is checked whether or not it has any child object (step S213). If the object has any child object, then control is returned to the process in step S212 to retrieve the child object. If the object has no child objects, it is checked whether or not the object has any younger brother object (step S214). If yes, control is returned to the process in step S212 to retrieve the younger brother object. If the object has no younger brother objects, its parent object is to be retrieved, but is not actually retrieved (step S215). Then, it is checked whether or not the parent object is a root object (step S216). If yes, the retrieval is determined to have failed. Unless the parent object is a root object, the processes in and after step S214 are performed. Thus, the CG data are retrieved.

The above described operation of the drive simulator is explained below furthermore in detail. For example, the retrieved three-dimensional coordinate of the viewpoint, the movement direction vector, and the speed are set as follows.

3-dimensional coordinate of viewpoint:
(100. 0, 200. 0, 0. 6)
movement direction vector:
(1. 0, 0. 0, 0. 0)
speed: 10.0

The movement direction vector is extended toward the positive direction of the x axis.

The CG calculating unit 192 changes the direction vector by +3° to the right based on the data received from the CG data referring unit 193 and the data received from the input interpreting unit 191 indicating a 3-picture-element movement to the right. That is, the changed movement direction vector is represented as follows.

movement direction vector:
(0. 998, 0. 052, 0. 0)

Then, the CG calculating unit 192 calculates the increment after the movement based on a new movement direction vector and speed. At this time, a velocity vector is generated by multiplying each element of the new movement direction vector by the speed. The resultant velocity vector is represented as follows.

velocity vector: (9. 980, 0. 520, 0. 0)

The velocity vector gives a movement increment for the viewpoint.

Furthermore, the CG calculating unit 192 calculates the three-dimensional coordinate of a new viewpoint by adding each element of the velocity vector to the three-dimensional coordinate of the present viewpoint. The three-dimensional coordinate is represented as follows.

three-dimensional coordinate of the viewpoint:
(109,. 980, 200. 520, 0. 6)

Then, the CG calculating unit 192 requests the CG data storing unit 195 to store the three-dimensional coordinate of

	Document ID	U	Title	Current OR
82	US 5766212 A	☒	Disposable diaper	604/361
83	US 5728492 A	☒	Mask for projection system using charged particle beam	430/5
84	US 5727064 A	☒	Cryptographic system for wireless communications	380/270
85	US 5718991 A	☒	Method for making photomasks having regions of different light transmissivities	430/5
86	US 5689117 A	☒	Apparatus for image transfer with charged particle beam, and deflector and mask used with such apparatus	250/492.23
87	US 5657754 A	☒	Apparatus for non-invasive analyses of biological compounds	600/316
88	US 5631750 A	☒	Scattering type liquid crystal device	349/110
89	US 5610705 A	☒	Doppler velocimeter	356/28.5
90	US 5598410 A	☒	Method and apparatus for accelerated packet processing	370/469
91	US 5557105 A	☒	Pattern inspection apparatus and electron beam apparatus	250/310
92	US 5534893 A	☒	Method and apparatus for using stylus-tablet input in a computer system	345/179
93	US 5532496 A	☒	Proximity effect compensation in scattering-mask lithographic projection systems and apparatus therefore	250/492.22
94	US 5517660 A	☒	Read-write buffer for gathering write requests and resolving read conflicts based on a generated byte mask code	711/117
95	US 5512759 A	☒	Condenser for illuminating a ringfield camera with synchrotron emission light	250/492.1
96	US 5506359 A	☒	Cocaine analogues and their use as cocaine drug therapies and therapeutic and imaging agents for neurodegenerative disorders	546/130
97	US 5502306 A	☒	Electron beam inspection system and method	250/310
98	US 5500312 A	☒	Masks with low stress multilayer films and a process for controlling the stress of multilayer films	430/5
99	US 5486919 A	☒	Inspection method and apparatus for inspecting a particle, if any, on a substrate having a pattern	356/484
100	US 5471628 A	☒	Multi-function permutation switch for rotating and manipulating an order of bits of an input data byte in either cyclic or non-cyclic mode	712/223
101	US 5439781 A	☒	Device fabrication entailing synchrotron radiation	430/311
102	US 5438405 A	☒	Device and method for testing optical elements	356/239.2
103	US 5430292 A	☒	Pattern inspection apparatus and electron beam apparatus	250/310
104	US 5384463 A	☒	Pattern inspection apparatus and electron beam apparatus	250/398

the new viewpoint and the movement direction vector. The CG data storing unit **195** stores these data in the CG data storage unit **162**. The stored three-dimensional coordinate of the viewpoint and the movement direction vector are represented as follows.

three-dimensional coordinate of the viewpoint:
(109. 980, 200. 520, 0. 6)
movement direction vector:
(0. 980, 0. 052, 0. 0)

After the CG data storing unit **195** has stored the data, the CG calculating unit **192** requests the change data storing unit **194** to store change data. The change data refer to the three-dimensional coordinate of a new viewpoint. The three-dimensional coordinate of the three-dimensional viewpoint stored by the change data storing unit **194** is acquired by the image generating unit **202** of the second processing unit **23**. That is, the image generating unit **202** reads the three-dimensional coordinate of the viewpoint stored in the changed data buffer **161**.

three-dimensional coordinate of the viewpoint:
(109. 980, 200. 520, 0. 6)

Then, the image generating unit **202** transmits the three-dimensional coordinate of the viewpoint to the change data referring unit **203** and requests it to update the data. By the instruction from the change data referring unit **203**, the changed CG data storing unit **204** updates and stores the three-dimensional coordinate of the viewpoint stored in the changed CG data storage unit **164**.

Then, the image generating unit **202** refers to the CG data to be displayed through the changed CG data referring unit **201**, and generates images. The changed CG data referring unit **201** refers to the changed CG data storage unit **164** using a pointer, and passes the reference result to the image generating unit **202** which generates and displays images. FIG. **66B** shows the new image to be displayed. Since the movement direction vector is vertical to the screen and is extended inward the screen as in the case shown in FIG. **66A**, the x axis is turned 3° to the left compared with the state shown in FIG. **66A**. Accordingly, the road surface is represented and displayed as being entirely changed.

FIG. **69** shows the movement of a viewpoint. When a chair is displayed in a state ST on a screen, clicking a mouse cursor X at a left point to the center of the screen moves the viewpoint to the left, thereby displaying the chair in a state ST' from a different viewpoint.

FIG. **70** shows an example of the configuration of the CG data according to the fourth embodiment. In the example according to the fourth embodiment, the CG data stored in the CG data storage unit **162** are represented by process codes. For example, when a chair is represented, the chair is specified using a process code through a root object. Then, a back object and a bottom object are provided as child objects of the chair. The back and bottom objects have information about their forms for use in display. A camera position object is stored at a brother position, that is, at the equal level as that of the chair. That is, the objects are stored in a tree structure.

According to the fourth embodiment, the storage format can be either the same or different between the CG data storage unit **162** and the changed CG data storage unit **164**. For example, as shown in FIG. **70**, the CG data can be stored at a same level in the changed CG data storage unit **164**. That is, the items such as a back form, bottom form, camera position, etc. are defined as data at an equal level, thereby equalizing the access time in a display process.

FIG. **71** shows an example of another configuration according to the fourth embodiment. In FIG. **71**, a unit also shown in FIG. **54** is assigned a corresponding number. According to the embodiment shown in FIG. **54**, data are transmitted through the shared memory **178**. According to the embodiment shown in FIG. **71**, data are transmitted through a communication circuit **241** of a first processor **244** and a communication circuit **242** of a second processor **245**. A buffer **243** is provided between the communication circuits to release the first processor **244** and the second processor **245** from a synchronous operation with each other. The buffer **243** corresponds to the change data buffer **161** shown in FIG. **53**.

The CG data display device is described above according to the fourth embodiment, but the fourth embodiment is not limited to a system operated with a multiprocessor. For example, it can be operated with a single-processor system when a multiple-process function is provided under the control of an operating system. The communications between a plurality of processes are performed through an inter-process communication, for example, like a socket in a UNIX system. FIG. **72** shows the CG data display device according to the fourth embodiment realized with a UNIX system. In FIG. **72**, a first process **251** performs a CG data management process shown in FIG. **57**, and transmits change data to a second process **252** through a socket connected between the first process **251** and the second process **252**. The first process **251** writes change data onto the socket in the change data storing process in step S179 shown in FIG. **57**. The second process **252** performs the CG data image generating process shown in FIG. **60**, and reads change data from the socket in the change data referring process in step S201 shown in FIG. **61**. In the fourth embodiment, since the amount of data communicated through the socket is reduced to the lowest possible level, the image generating process can be performed at a high speed.

Furthermore, the first processing unit **18** and the second processing unit **23** shown in FIG. **53** can be designed separately as a logic circuit. According to the embodiment shown in FIG. **54**, external CG data are stored in a RAM. If display contents are limited, however, they can be preliminarily stored in a ROM and transmitted to a RAM when necessary.

As described above, a process performed in a CG data display device can be divided into two portions for each of which a processing unit is provided according to the fourth embodiment. Since each processing unit is provided with CG data, only change data relative to a change has to be transmitted, thereby shortening the time taken for transmission and successfully realizing a high-speed process. Furthermore, the two processing units operate in parallel and the amount of data transmitted between the two processes are considerably reduced. Therefore, the image display process can be performed at a high speed with dynamic images represented realistically and static images changed immediately after a change of scene.

FIG. **73** shows the configuration of the important portion of the three-dimensional object display device according to the fifth embodiment of the CG data display device of the present invention. The fifth embodiment comprises an object data structure compressing unit **263** and a display-format hierarchical object data structure storage unit **264** not included in the conventional device shown in FIG. **1**. As

	Document ID	U	Title	Current OR
105	US 5331446 A	☒	Liquid crystal optical element and a laser projection apparatus using polymer dispersed liquid crystal	349/5
106	US 5309273 A	☒	Yag laser mask marker	359/202
107	US 5298969 A	☒	Combined optical train for laser spectroscopy	356/340
108	US 5298968 A	☒	Combined optical train for laser spectroscopy	356/338
109	US 5282151 A	☒	Submicron diameter particle detection utilizing high density array	702/26
110	US 5224214 A	☒	BuIffet for gathering write requests and resolving read conflicts by matching read and write requests	710/39
111	US 5162645 A	☒	Photographic scanner with reduced susceptibility to scattering	250/208.1
112	US 5123095 A	☒	Integrated scalar and vector processors with vector addressing by the scalar processor	712/218
113	US 5099117 A	☒	Scanning tunnel microscope capable of detecting electrons emanating from a specimen	250/306
114	US 5028135 A	☒	Combined high spatial resolution and high total intensity selection optical train for laser spectroscopy	356/340
115	US 5017016 A	☒	Method of processing asbestos chips and apparatus	366/139
116	US 4924254 A	☒	Film printing/reading system	355/20
117	US 4922115 A	☒	Fluorescent glass dosimeter	250/484.5
118	US 4912022 A	☒	Method for sloping the profile of an opening in resist	430/396
119	US 4886974 A	☒	Mark detecting device for detecting the center of a mark by detecting its edges	250/559.36
120	US 4814626 A	☒	Method for high precision position measurement of two-dimensional structures	250/559.3
121	US 4812620 A	☒	Concentrated radiant energy heat source unit	392/421
122	US 4796997 A	☒	Method and system for high-speed, 3-D imaging of an object at a vision station	356/608
123	US 4776693 A	☒	Foreign substance inspecting system including a calibration standard	356/237.3
124	US 4771470 A	☒	Noise reduction method and apparatus for medical ultrasound	382/266
125	US 4764776 A	☒	Thermo transfer printer	347/232
126	US 4764441 A	☒	Photo-mask for production of substrate for optical memory element	430/5
127	US 4764013 A	☒	Interferometric apparatus and method for detection and characterization of particles using light scattered therefrom	356/484

shown in FIG. 73, the object data structure compressing unit 263 is connected between an editing-format hierarchical object data structure storage unit 262 and a display-format hierarchical object data structure storage unit 264. The display-format hierarchical object data structure storage unit 264 outputs display CG data to an object display process unit 265.

A model instruction is transmitted from a user to an object generating/editing unit 261, and a display instruction is concurrently transmitted from the user to the object display process unit 265. In FIG. 73, there are two storage units each having a hierarchical object data structure. One unit has an editing-format hierarchical object data structure, and the other has a display-format hierarchical object data structure.

Upon receipt of a model instruction from a user, the object generating/editing unit 261 updates the hierarchical object data structure in the editing-format hierarchical object data structure storage unit 262. At this time, the object data structure compressing unit 263 compresses the hierarchical object data structure according to predetermined rules, and updates the display-format hierarchical object data structure in the display-format hierarchical object data structure storage unit 264 based on the compression result. The object display process unit 265 processes the data in the display-format hierarchical object data structure storage unit 264, generates image data, and displays them on the screen of the display device 4.

Thus, the display-format hierarchical object data structure are sequentially updated, compressed, retrieved at a user's display instruction, and actually displayed. The display-format hierarchical object data structure is considerably reduced in depth of structure or in number of objects when compared with the editing-format hierarchical object data structure in the editing-format hierarchical object data structure storage unit 262, thereby shortening retrieval time and ensuring a high-speed process.

The process of compressing the hierarchical object data structure according to the fifth embodiment is very unique. That is, when the hierarchical object data structure in the editing-format hierarchical object data structure storage unit 262 is updated through the object generating/editing unit 261 at a user's model instruction, the object data structure compressing unit 263 compresses the editing-hierarchical object data structure, and simultaneously updates the display-format hierarchical object data structure based on the compression result.

Compressing a hierarchical object data structure is expanding a child object inheriting an attribute of a parent object and storing the child object in the parent object. The expanding process copies the attribute of the child object to its parent object. However, this process should be performed by checking a user's access request so as not to affect an operation in a user's modeling process.

The process of compressing a hierarchical object data structure presents a problem when objects are simply combined. That is, the attribute inheritance characteristic between a parent object and a child object should be maintained. For example, FIG. 74B shows an example of an erroneous compression in which the inheritance of objects shown in FIG. 74A is ignored. In FIG. 74A, although a child object having form β has been generated such that it inherits color A of its parent, it actually shows color B of a child object having form α as a result of the compression shown in FIG. 74B.

Typical attributes having inheritance characteristics of an object are color, texture, and transformation matrix. The attribute not having the inheritance characteristics is form.

A number of objects use the inheritance characteristics for color and texture. That is, utilizing the inheritance characteristics for color and texture can considerably reduce the number of objects and hierarchical levels.

Regarding the transformation matrix, an object data structure can be compressed under a consideration of inheritance of it as well as color and texture if the matrix relates to a world coordinate which is an absolute coordinate. If the transformation matrix of a child is a relative transformation matrix between the coordinate of a parent and that of a child, then the transformation matrix in a world coordinate system of the child object is calculated by multiplying the transformation matrix of the world coordinate system of the parent by the relative transformation matrix. Therefore, a relative transformation matrix does not show inheritance of an attribute. In this case, the transformation matrix of the world coordinate system of the child is also calculated and stored in the display-format hierarchical object data structure storage unit 264 to perform a compressing process.

FIG. 75 shows an example of a compressing process for a color attribute of an object shown in FIG. 2. The process is performed by the object data structure compressing unit 263 shown in FIG. 73. FIG. 2 shows the hierarchical object data structure at the time of editing an object. FIG. 75 shows the hierarchical object data structure after the compressing process according to the fifth embodiment. In this case, an object data structure compressing process is performed using the attribute inheritance characteristic. The present example is also an object-editing-format hierarchical object data structure as a model of the room represented by the conventional method shown in FIG. 2.

As shown in FIG. 75, the four objects as the feet of the chair inherit color B of the parent legs. Accordingly, these objects can be ended into the parent and stored, and they maintains the inheritance characteristic after expansion. Since the bottom panel and the back panel similarly inherit color B of the parent indicating the chair, they can be compressed and expanded into the parent object. A "expansion" indicates that an attribute of a child node is copied to its parent node. However, the legs object does not inherit the color of the chair, that is, the parent of the legs, but indicates its own color E. Therefore, it cannot be expanded. FIG. 75 shows the final display-format hierarchical object data structure processed by the compression through the above described expansion.

In FIG. 2, the depth of the hierarchy of the object tree is four levels and the number of objects is 11. On the other hand, in FIG. 75, the depth of the hierarchy is 3 levels and the number of objects is 5. That is, the object data structure is compressed.

FIG. 76B shows an example of a compressing process for the attribute of the transformation matrix of the object shown in FIG. 76A. FIG. 76A shows the hierarchical data structure at a time of editing an object. FIG. 76B shows the hierarchical object data structure after the compressing process according to the fifth embodiment. FIG. 76A shows the case in which four child objects having specific colors and forms inherit the transformation matrix M2 of their parent. In this case, simply changing the transformation matrix M2 of the parent converts the coordinates of the four child objects according to the changed transformation matrix. The child objects can be expanded into their parent object and stored, thereby the object data structure can be compressed.

In FIG. 76A, the depth of the hierarchy is 3 levels, and the number of objects is 6. On the other hand, in FIG. 76B, the depth of the hierarchy is 2 levels, and the number of objects is 2 by compressing the object data structure.

	Document ID	U	Title	Current OR
128	US 4710025 A	☒	Process for characterizing suspensions of small particles	356/343
129	US 4634876 A	☒	Object position detecting apparatus using accumulation type sensor	250/548
130	US 4548500 A	☒	Process and apparatus for identifying or characterizing small particles	356/336
131	US 4457893 A	☒	Automated apparatus for photometrically detecting immunological agglutinating reactions	422/64
132	US 4417946 A	☒	Method of making mask for structuring surface areas	216/2
133	US 4392236 A	☒	System and method of migratory animal identification by fluorescence spectroscopy of element coded implanted tags, and tags used therein	378/45
134	US 4342817 A	☒	Mask for structuring surface areas, and method of making it	430/5
135	US 4325910 A	☒	Automated multiple-purpose chemical-analysis apparatus	422/64
136	US 4146883 A	☒	Display	340/815.44
137	US 4050638 A	☒	Radioactive matter containing waste gas treating installation	241/222
138	US 3972598 A	☒	Multifaceted mirror structure for infrared radiation detector	359/853
139	US 3936694 A	☒	Display structure having light emitting diodes	313/500
140	US 3905675 A	☒	Optical systems having stop means for preventing passage of boundary wave radiation	359/434
141	US 3873204 A	☒	Optical extinction photoanalysis apparatus for small particles	356/39
142	US 3744878 A	☒	LIQUID CRYSTAL MATRIX WITH CONTRAST ENHANCEMENT	349/177
143	US 3713743 A	☒	FORWARD SCATTER OPTICAL TURBIDIMETER APPARATUS	356/338
144	US 3614231 A	☒	OPTICAL AEROSOL COUNTER	356/37

FIG. 77B shows an example of a compressing process for the attribute of the object having the relative transformation matrix shown in FIG. 77A. FIG. 77A shows the hierarchical data structure at a time of editing an object. FIG. 77B shows the hierarchical object data structure after the compressing process according to the fifth embodiment. Fundamentally, the compressing process is performed similarly as shown in FIG. 75 while the relative transformation matrix R1 in the object-editing-format hierarchical object data structure is replaced with the transformation matrix in the world coordinate system (world transformation matrix) W1. Then, the world transformation matrix W1 is stored in the display-format hierarchical object data structure.

FIG. 77A shows the case in which four child objects having specific colors and forms inherit the relative transformation matrix R1 of their parent. Simply changing the relative transformation matrix R1 of the parent converts the coordinates of the four child objects. In this case, simultaneously calculating the transformation matrices of the world coordinate system of the parent and the four children according to the method described later expands and stores the child objects to the parent object with respect to the world transformation matrices. Thus, the object data structure can be compressed.

FIG. 78 shows the contents of each node in the object tree structure. As shown in FIG. 78, the contents are a pointer to the parent node, color information, texture information, matrix information, number of children, and pointers to the children. If a node inherits the color, texture, and matrix of its parent node, the node does not contain the information.

FIG. 79 is the flowchart of the processes performed by the object data structure compressing unit 263 shown in FIG. 73. If retrieval is started from the top node (root) of an object data structure (step S221), then it is determined whether or not there are any nodes to be processed (step S222). If no, the process terminates. If yes in step S224, the attribute information of the node is acquired (step S223). Then, it is determined whether or not there are any child nodes to the node (step S224). If no, it is determined whether or not a compressing process can be performed (step S227). If yes in step S224, the attribute information of the child node is acquired (step S225). Then, it is determined whether or not an attribute is inherited (step S226). If no, it is determined whether or not a compressing process can be performed (step S227). If yes in step S226, control is returned to the process in step S224.

In the above described step S226, it is estimated whether or not attributes are inherited with respect to all attributes such as color, texture, matrix, etc. If a matrix remains to be inherited after the inheritance of color is processed, then a child node at a lower level is inquired of the existence of the inheritance of a matrix. A child node inheriting the attribute is sequentially stored as a child node to be processed in a compressing process. If a user specifies an attribute to be compressed, the existence of inheritance characteristic is checked only for the specified attribute.

If a compressing process cannot be performed in step S227 because, for example, a child node does not inherit an attribute, then the child node is a non-inheritance node, and the processes are performed from step S222 again. If the compressing process can be performed, the compressing process is carried out (step S228). The inheritance attribute remaining in the child node at this time is the longest inheritance characteristic. Then, the stored child node is compressed with respect to the attribute. In the compressing process, all attribute elements of a child node which inherits attributes are expanded below attribute elements of a specified node (a parent node). Then, the expanded child node is deleted. If the attribute-inheriting child node is further defined below the child node, then all attribute elements are expanded to the specified node and the child node is deleted in the same manner. Thus, child nodes are expanded to a specified node and the child nodes are deleted, thereby reducing the number of levels of an object data structure and the entire number of nodes.

If the compressing process has been completed in step S228, then control is passed to another unretrieved node (step S229).

In this case, the unretrieved node closest to the top node is defined as a specified node. To process the inheritance of an attribute, the object tree should be retrieved from an upper level.

The process shown in FIG. 79 is explained in detail using an example of the object tree shown in FIG. 80. Each node of the object tree shown in FIG. 80 has the data structure as shown in FIG. 78.

The number of the children of top node N0 is 4. Since top node N0 has the child nodes (yes in step S224), the attribute information about the first child node, that is, child node N1, pointed to by pointer PTR1 is acquired (step S225). Then, a compressing process is performed on child node N1 (step S228). Since child node N1 has no child nodes, control is returned to top node NO at the highest level (step S229).

Then, child node N2 pointed to by the next pointer PTR2 is processed (step S225). Since child node N2 has child node N5 pointed to by pointer PTR5, child node N5 is processed (step S225), and a compressing process is performed on child nodes N2 and N5 (step S228). Since child node N5 has no child nodes, control is returned to child node N2 at a higher level. Since the only one child node N5 of child node N2 has been processed, control is returned to top node NO at the highest level (step S229).

Thus, the nodes are recursively retrieved in a compressing process, and child nodes N3, N4 are accessed using child node pointers PTR3 and PRT4 respectively in a compressing process.

FIG. 81 shows an example to explain a world transformation matrix. In this example, a square desk is placed in a room as shown in FIG. 81. A world coordinate system (absolute coordinate system) is used to define the form of the desk. In this system, one corner of the room is defined as the origin O. The center of the top panel 271 of the desk is assigned the position (x, y, z)=(100, 100, 45). The size of the top panel 271 of the desk is 100 in width, 100 in depth, and 10 in thickness. The four legs 272, 273, 274, and 275 of the desk are circular cylinders with 5 in radius and 40 in height, and separately fixed to the four corners of the top panel. At this time, a XY plane corresponds to the floor of the room.

If the desk is modeled using the world coordinate system under the above described condition, the coordinates of eight vertices of a rectangular parallelepiped of the top panel 271 and the coordinates of the centers of bottom faces of the circular cylinder legs are obtained as shown in FIG. 81.

FIG. 82 shows a modeling coordinate system of the top panel 271. FIG. 83 shows a modeling coordinate system of the legs 272, 273, 274, and 275. As shown in FIG. 82, a local coordinate system is defined with the center of the top panel 271 of the desk set as the origin O.

A modeling coordinate system is a coordinate system for unique use in defining the form of an object. That is, the form of an object can be defined without considering the position in the world coordinate system, thereby allowing a user to immediately model the object.

Likewise, a modeling coordinate system can be defined as shown in FIG. 83 for a leg of the desk with the center of a

	Document ID	U	Title	Current OR
1	JP 2003050207 A	☐	SENSOR FOR PRINT INSPECTION	
2	JP 2002096461 A	☒	DEVICE FOR IMAGE RECORDING, METHOD FOR CONTROLLING IMAGE RECORDING, AND RECORDING MEDIUM	
3	JP 2000191992 A	☒	MASKING TAPE AND METHOD FOR MASKING USING THE TAPE	
4	JP 2000100674 A	☒	METHOD FOR MARKING SEMICONDUCTOR WAFER	
5	JP 11179962 A	☒	ELECTROOPTICAL SIGNAL CONVERTING APPARATUS	
6	JP 10193145 A	☒	LASER MARKING DEVICE	
7	JP 10096700 A	☒	APPARATUS FOR INSPECTING FOREIGN MATTER	
8	JP 09106065 A	☒	SUBSTRATE CLEANING DEVICE AND METHOD	
9	JP 07198625 A	☒	PRINT INSPECTING SENSOR	
10	JP 07020793 A	☒	PRODUCTION OF BLACK MASK FILTER FOR LED DISPLAY BY SCREEN PRINTING	
11	JP 05343808 A	☒	MANUFACTURE OF OPTICAL SEMICONDUCTOR ELEMENT	
12	JP 05241011 A	☒	PRODUCTION OF COLOR FILTER FOR LIQUID CRYSTAL DISPLAY	
13	JP 05079913 A	☒	STRAY LIGHT FREE FOURIER SPECTROPHOTOMETER	
14	JP 04194908 A	☒	LIQUID CRYSTAL DISPLAY DEVICE	
15	JP 03017692 A	☒	COLOR DISPLAY DEVICE	
16	JP 02271644 A	☒	CARRYING DEVICE OF GLASS SUBSTRATE	
17	JP 01259244 A	☒	FOREIGN MATTER DETECTION SYSTEM	
18	JP 01173891 A	☒	FLUORESCENT GLASS DOSIMETER	
19	JP 01096601 A	☒	DEFECT CORRECTING METHOD FOR COLOR FILTER	
20	JP 63070110 A	☒	DISTANCE MEASURING APPARATUS	
21	JP 60024568 A	☒	COLOR TONER CONCENTRATION DETECTOR	

circular cylinder of the leg set as the origin O. After the definition, the modeling coordinate systems are converted into the world coordinate system using the world transformation matrices of the top panel and the legs, thereby generating a desk in the room. In this example, the world transformation matrix is simple and has an amount of a displacement as an element.

FIG. **84** shows a world transformation matrix which transforms the top panel **271** shown in FIG. **82** to the world coordinate system shown in FIG. **81**, and a world transformation matrix which transforms the leg shown in FIG. **83** to the position of the leg **272** shown in FIG. **81**. Each world transformation matrix shown in FIG. **84** is one of the well-known affine transformation matrices, and indicates a parallel displacement of a point in a three-dimensional space. Generally, a point (x, y, z) in an XYZ space is moved to the point (x', y', z') represented by the following equation using the three-dimensional affine transformation matrix M_A.

$$(x',y',z',1)=(x,y,z,1)M_A \tag{12}$$

FIG. **85** shows various three-dimensional affine transformation matrices. The transformation matrix M_P is a transformation matrix through which the point (x, y, z) is moved in parallel to the X, Y, and Z axes by T_x, T_y, and T_z respectively. The transformation matrix M_s is a transformation matrix through which the coordinate values of the point (x, y, z) are multiplied by S_x, S_y, and S_z respectively, and indicates scale conversion including enlargement, reduction, and inversion. Using the transformation matrices M_{RX}, M_{RY}, and M_{RZ}, the point (x, y, z) is rotated by θ degrees around the X, Y, and Z axes respectively.

The transformation matrix through which a three-dimensional object is transformed is used as follows. First, the transformation matrices M_{RX}, M_{RY}, and M_{RZ} produce rotations such that each axis of coordinate of a modeling coordinate system of a polyhedron to be modified can match a corresponding axis of coordinate in the modeling coordinate system of a specified polyhedron. Then, a parallel displacement is applied to the polyhedron using the transformation matrix M_P.

If user-selected lines are to be overlapped, a rotation and a parallel displacement are carried out so that the lines are overlapped. If it is necessary to set the lines in the same length, then a scale conversion is conducted using the transformation matrix M_s so that a length of a line to be modified matches a length of a specified line.

If selected points should be overlapped, only the transformation matrix M_P is used to conduct a parallel displacement. The transformation matrices M_P, M_S, M_{RX}, M_{RY} and M_{RZ} are multiplied to the coordinates of each vertex of a polyhedron to be modified in the above described manner. Thus, the polyhedron is transformed in a three-dimensional space.

Each world transformation matrix shown in FIG. **84** corresponds to the transformation matrix M_P shown in FIG. **85**. The center of the top panel **271** shown in FIG. **82** is moved from the origin to the point (100, 100, 45) according to the world transformation matrix of the top panel. The center of the leg shown in FIG. **83** is moved from the origin to the point (145, 55, 20) according to the world transformation matrix of the leg. The relationship between the world coordinate system and the modeling coordinate system can be defined by an arbitrary affine transform other than a parallel displacement if necessary. Then, a necessary change associated with the movement of a form can be made using the world transformation matrix, and each coordinate of the form need not be changed at all.

FIG. **86** shows relative transformation matrices of the legs **272**, **273**, **274**, and **275** generated with the center of the top panel **271** as a base point. For example, according to the relative transformation matrix of the leg **272**, the center (0, 0, 0) of the top panel shown in FIG. **80** is moved to the point (45, −45, −25) indicating the relative position of the center of the leg **272**. Since the relative transformation matrices are generated in the modeling coordinate system of the top panel **271**, the legs can be modeled without considering the position of the top panel **271** in the room. Furthermore, since the relative transformation matrices are assigned regardless of the position of the top panel **271**, displacement and rotation can be produced even for the legs by changing the world transformation matrix of the top panel **271** in the world coordinate system. Therefore, a user can be provided with an efficient modeling environment. Actually, since the position of each leg in the world coordinate system is required to display the forms of the legs, the world transformation matrix of each leg is obtained by multiplying the world transformation matrix of the top panel **271** by the relative transformation matrix of each leg. However, the user need not be informed of the world transformation matrix of each leg obtained by the multiplication.

FIG. **87** shows an example of the hierarchical object data structure relating to the relative transformation matrix stored in the editing-format hierarchical object data structure storage unit **262**. In FIG. **87**, the relative transformation matrix of the top panel is generated with the corner of the room set as an base point. Since the corner of the room is the origin of the world coordinate system, it is also a world transformation matrix of the top panel. The relative transformation matrices of the legs are defined with the center of the top panel set as an base point. In FIG. **87**, the color and texture are omitted, and only the relative transformation matrix is considered. Since the relative transformation matrices of the legs are defined relative to the position of the top panel, they are represented by a parent-child relationship between the top panel and the legs.

FIGS. **88** and **89** show examples of hierarchical object data structures relating to a world transformation matrix. In FIGS. **88** and **89**, the color and texture are omitted, and only the relative transformation matrix is considered as an attribute of each object. The world transformation matrix is, unlike the relative transformation matrix, generated based on the origin of the world coordinate system. Therefore, a number of combinations of a parent and children among objects can be defined. For example, in FIG. **88**, four legs are the child objects of the top panel. However, each leg does not have to be necessarily the child of the top panel. As shown in FIG. **89**, it can be a child object of the desk which is an object at a higher level.

The relative transformation matrix is not required as an attribute in the display-format hierarchical object data structure as long as the world transformation matrix is available. The world transformation matrix of a child object can be used in a display process independently of the world transformation matrix of a parent object. Then, each relative transformation matrix of the editing-format hierarchical object data structure shown in FIG. **87** is transformed into the world transformation matrix so as to generate the hierarchical object data structure shown in FIG. **88** or **89**, thereby it can be compressed.

FIG. **90** shows an example of a structure obtained after compressing the hierarchical object data structure shown in FIG. **88**. In FIG. **88** showing the state before the compression, the depth of the hierarchy is 3, and the number of objects is 6. On the other hand, in FIG. **90** showing the

	Document ID	U	Title	Current OR
22	JP 59054040 A	☒	OPTICAL HEAD	
23	JP 58038082 A	☒	INFRARED-RAY IMAGE PICKUP DEVICE	
24	JP 57135404 A	☒	DISK DEFECT INSPECTING DEVICE	
25	JP 57130471 A	☒	MANUFACTURE OF METAL OXIDE SEMICONDUCTOR FIELD-EFFECT TRANSISTOR	
26	DE 10127689 A1	☒	Generating integrated electrical circuit manufacturing mask structure scatter bars involves generating bars near edges or between edge pairs, checking and correcting separations	
27	WO 9932921 A1	☒	DISPLAY DEVICE FOR PROJECTOR AND METHOD OF MAKING AND USING A DISPLAY DEVICE	
28	WO 9821629 A2	☒	IN-LINE HOLOGRAPHIC MASK FOR MICROMACHINING	
29	WO 9605503 A1	☒	DEVICE FOR TESTING OPTICAL ELEMENTS	
30	JP 2003114183 A	☒	Production of light generation element for use in e.g. light microscope, involves forming scattering object mask on light propagation object, with which aperture is formed to hold micro scattering object	
31	DE 10127689 A	☒	Generating integrated electrical circuit manufacturing mask structure scatter bars involves generating bars near edges or between edge pairs, checking and correcting separations	
32	US 2002002149 2 A	☒	Stereoscopic image display method in TV, involves guiding display light from specific strip images of parallax image to observation position, through mask using lenticular lenses	
33	EP 1065566 A	☒	Electron beam drawing mask blank for integrated circuit, includes pattern supporting layer, electron beam scattering layer, etching stopper layer and support layer formed of preset element with preset film thickness	
34	WO 200054097 A	☒	Active electro-optic filtering device for use with a welder's protection mask reduces light scattered from LCD filter element and has reduced operating voltage to give improved optical quality	
35	JP 11179962 A	☒	Electric-light signal converter for video printer - has mask member fixed to light emitting element holder, for cutting off scattered light	
36	US 5866913 A	☒	Proximity correction dose modulation for E-beam projection lithography - using a pattern defining mask containing sub-resolution scattering features	
37	US 5844722 A	☒	Polarization beam splitter for colour projection system - has wave blocking element arranged at bottom edge of mask and immersed in prismatic fluid, for minimizing scattering of incident electromagnetic wave	
38	US 5718991 A	☒	Multi-exposure photomask preparation with at least three levels of transmission - by forming high exposure photomask with a number of through holes and masked area, then forming a refractive light scattering optical element above it	
39	JP 09257685 A	☒	Photodetector for measuring particle size distribution in specimen - includes mask on light receiving surface to focus scattered light into fixed area of light receiving element	
40	JP 06148652 A	☒	Liq. crystal display element with improved display quality - has resin layer mixed with light scattering microparticles provided on substrate at picture-element-free region	
41	EP 556655 A	☒	Grading and evaluating method for optical elements such as lenses - scanning rotated linear wedge shaped beam of white light on entire lens surface and detecting defect scattered light using CCD via mask	

state after the compression, the depth of the hierarchy is 2, and the number of the objects is 2. Therefore, objects can be retrieved at a higher speed in a display process by using the hierarchical object data structure shown in FIG. 90.

FIG. 91 is a flowchart of the compressing process performed by the object data structure compressing unit 263 when the editing-format hierarchical object data structure contains as an attribute the matrix information such as a relative transformation matrix, a world transformation matrix, etc.

In FIG. 91, a compressing process is started from the top node (step S231), a child node is retrieved (step S232), and it is determined whether or not a child node has a relative transformation matrix as an attribute (step S233).

If a child node has a relative transformation matrix, then the world transformation matrix of the child is obtained by multiplying the world transformation matrix of the parent by the relative transformation matrix (step S234). Next, the obtained world transformation matrix of the child is copied to the parent node (step S235), the child node is deleted (step S236), and the processes in and after step S232 are repeated.

If the child node has no relative transformation matrix in step S233, then the processes in and after step S235 are performed. Then, in step S232, the process terminates if there are no child nodes to be compressed.

Thus, the compressed hierarchical object data structure are stored in the display-format hierarchical object data structure storage unit 264, and used by the object display process unit 265 in generating image data.

As described above, according to the fifth embodiment, a compressing process is performed while maintaining an effective attribute inheritance characteristic in a modeling process. Simultaneously, a display-format hierarchical object data structure is concurrently updated. Thus, an object retrieval process can be reduced, thereby considerably shortening the time taken for a displaying process and realizing a high speed display.

What is claimed is:

1. A computer graphics data display device for displaying an object in a computer graphics world comprising:

attribute defining/setting means for defining and setting an attribute indicating a type of calculation to determine a state change for at least one object in the computer graphics world;

attribute memory means for storing the attribute of the at least one object set by said attribute defining/setting means;

state change calculating means for calculating the state change of a displayed object according to the attribute of at least one related object in response to one of a movement of the displayed object in the computer graphics world and receipt of an external activation instruction, the at least one related object including one of the displayed object and an other object; and

result display means for displaying a result of a calculation.

2. The computer graphics data display device according to claim 1, further comprising:

viewpoint volume defining means for defining a volume of a viewpoint corresponding to the displayed object in the computer graphics world, wherein

said state change calculating means calculates a moved-to position of the viewpoint based on a relation between the volume of the viewpoint and the attribute of the other object.

3. The computer graphics data display device according to claim 1, wherein

said attribute defining/setting means defines an interference attribute as the attribute indicating the type of the calculation, the interference attribute relating to an interference between the displayed object and the object when the displayed object comes in contact with the other object; and

said state change calculating means performs a movement interference calculation for the displayed object using a calculation method specified by the interference attribute.

4. The computer graphics data display device according to claim 3, wherein

said interference attribute contains an interference restriction attribute indicating a constraint on the movement of the displayed object in the interference between the displayed object and the other object in the computer graphics world.

5. The computer graphics data display device according to claim 4, wherein

said attribute defining/setting means defines an average slope angle as the interference restriction attribute which is the constraint when the displayed object moves on the other object having the interference restriction attribute; and

said state change calculating means performs a movement calculation, based on the average slope angle, of the displayed object moving on the other object assigned the average slope angle.

6. The computer graphics data display device according to claim 4, wherein

said attribute defining/setting means defines an attribute indicating restriction information which depends on a level of a contact force generated by the interference as the interference restriction attribute; and

said state change calculating means calculates the contact force in the interference between the displayed object and the other object assigned the interference restriction attribute.

7. The computer graphics data display device according to claim 1, wherein

said attribute defining/setting means defines a reaction attribute as the attribute indicating the type of the calculation, the reaction attribute related to the displayed object which reacts with the other object when the displayed object comes in contact with the other object and starts the state change; and

said state change calculating means performs the movement interference calculation for the displayed object according to one of a trigger and a type of the state change of the displayed object specified by the reaction attribute.

8. The computer graphics data display device according to claim 7, further comprising:

sound output means for outputting a predetermined sound when the displayed object having a sound output attribute, indicating an output of a predetermined sound as the reaction attribute, comes in contact with the other object; wherein

said attribute defining/setting means defines the sound output attribute.

9. The computer graphic data display device according to claim 1, wherein

said attribute defining/setting means defines a reaction attribute as the attribute indicating the type of the calculation, the reaction attribute related to the dis-

	Document ID	U	Title	Current OR
42	JP 04197650 A	☒	Mfg. thermal head for printer - scattering sizes of heating-element portions of photo-mask, to prevent scattering in printing densities NoAbstract	
43	US 5028135 A	☒	Combined optical train for laser scattered light spectroscopy - uses pair of matched, novel optical elements, incorporating benefits of pinhole aperture and double lens, centre mask optical systems	
44	US 4912022 A	☒	Irradiating resist layer during manufacture of semiconductor devices - placing scattering element in path of radiation, modifying it to produce sloped edges in resist profile	

played object which reacts with the external activation instruction and starts the state change of one of the displayed object and other objects involved; and

said state change calculating means detects the external activation instruction and calculates the state change of the one of the displayed object and the other objects involved.

10. The computer graphics data display device according to claim 9, further comprising:

sound output means for outputting a predetermined sound when the activation instruction is issued to the displayed object having a sound output attribute indicating an output of a predetermined sound as the reaction attribute; wherein

said attribute defining/setting means defines the sound output attribute.

11. The computer graphics data display device according to claim 1, further comprising:

attribute distribution display means for displaying information about the attribute defined and set for the at least one object correspondingly to the at least one object in the computer graphics world.

12. A computer graphics data display method of displaying an object in a computer graphics world comprising the steps of: defining an attribute indicating a type of calculation to determine a state change for at least one object in the computer graphics world;

storing the defined attribute of the at least one object;

calculating the state change of a displayed object according to an attribute of at least one related object in response to one of movement of the displayed object in the computer graphics world and receipt of an external activation instruction, the at least one related object including one of the displayed object and an other object; and

displaying a calculation result.

13. A computer graphics data display device for use in a display system for displaying computer graphics data generated as graphic information by a computer, the computer graphics data display device comprising:

first processing means for detecting a change in first computer graphics data and, if there is a change detected in the first computer graphics data outputting second computer graphics data relating to the change as change data; and

second processing means for storing display computer graphics data corresponding to the first computer graphics data and used to generate image data, said second processing means receiving the change data output from said first processing means, updating the display computer graphics data using the change data, and generating the image data based on updated display computer graphics data.

14. The computer graphics data display device according to claim 13, further comprising:

change data buffer, provided between said first and second processing means, for storing the change data, wherein

said first and second processing means asynchronously performs respective processes.

15. The computer graphics data display device according to claim 13, wherein

said first processing means comprises:

computer graphics data memory means for storing the first computer graphics data; and

computer graphic data management means for obtaining the change data by calculating an influence of external information on the first computer graphics data stored in said computer graphics data memory means.

16. The computer graphics data display device according to claim 13, wherein

said second processing means comprises:

changed computer graphics data memory means for storing the display computer graphics data; and

computer graphics data image generating means for updating the display computer graphics data stored in said changed computer graphics data memory means and generating the image data from the display computer graphic data.

17. The computer graphics data display device according to claim 13, wherein

said first computer graphics data are output from a flight simulator comprising an operation integrating mechanism for obtaining information to be displayed based on data received from a pseudo flying device for a simulation of flying an airplane, calculated data of movement of an body of the airplane, and data indicating a flying method.

18. The computer graphics data display device according to claim 13, wherein

said computer graphics data are output from a run simulator comprising a run regenerating device for obtaining run display data to be displayed based on road surface form data designed according to a computer aided design, and run state data calculated from the road surface form data.

19. The computer graphics data display device according to claim 13,

said computer graphics data are output from a scene simulator comprising a scene generating mechanism for obtaining scene display data to be displayed based on form data designed according to a computer aided design and viewpoint data.

20. A computer graphics data display device for use in a display system for displaying computer graphics data generated as graphic information by a computer, comprising:

first processing means for outputting change data including first computer graphic data relating to a change if a change has arisen in second computer graphics data, said first processing means including:

a computer graphics data memory unit for storing the second computer graphics data; and

a computer graphics data management unit for obtaining the change data by calculating an influence of external information on the second computer graphics data stored in said computer graphics data memory unit;

second processing means including:

a changed computer graphics data memory unit for storing display computer graphics data corresponding to the second computer graphics data and used to generate image data; and

a computer graphics data image generating unit for updating the display computer graphics data stored in said changed computer graphics data memory unit based on the change data, and generating the image data from the display computer graphics data; and

a change data buffer, provided between said first and second processing means, for temporarily storing the change data.

21. The computer graphics data display device according to claim 20, wherein